

In theory, *You crack under pressure!*

Elvis Dohmatob

I. LIFE AT THE OPERA

“It’s opening night at the opera, and your friend is the prima donna (the lead female singer). You will not be in the audience, but you want to make sure she receives a standing ovation – with every audience member standing up and clapping their hands for her.” – Google code jam [1].

Notation: $\mathbb{N} = \{0, 1, 2, \dots\}$ denotes the natural numbers and $[k] := [0, k] \cap \mathbb{N} = \{0, 1, 2, \dots, k\}$, for every $k \in \mathbb{N}$.

Definition 1 (Shyness [1]). A spectator is said to have shyness level $s \in \mathbb{N}$ if they only stand up for an applause when s or more spectators are already up and applauding. Let $p_s \in \mathbb{N}$ be the number spectators with shyness level s .

Problem statement: Given a prescribed distribution (p_0, p_1, \dots, p_k) of shyness in the audience, with $p_k \neq 0$ (i.e. k is the shyness level of the shyest spectators), invite as few friends of the prima donna as possible (with their shyness levels distributed as you wish), say $r(p_0, p_1, \dots, p_k)$ friends, so that at the end she receives a standing ovation [1].

Remark 1. The natural number $r(p_0, p_1, \dots, p_k)$ is well-defined, thanks to the well-ordering principle.

A. Some examples to warm up.

The following examples are taken from the reference [1].

1. $r(1, 1, \dots, 1) = 0$. This is because the audience will eventually produce a standing ovation on its own.
2. $r(0, 9) = 1$. Inviting a *bold* friend is optimal.
3. $r(1, 1, 0, 0, 1, 1) = 2$. Inviting two friends with shyness level 2 is optimal.

II. SOLUTION: A SHORT RELIABLE PROGRAM

We will prove that: (a) $r(p_0, p_1, \dots, p_k) \leq k$; and (b) $r(p_0, p_1, \dots, p_k)$ is computable in linear time $\mathcal{O}(k)$. Moreover, the proof will be constructive, producing an algorithm which effectively computes $r(p_0, p_1, p_2, \dots, p_k)$ in k steps.

A. Preliminaries

Definition 2 (Insolubility of shyness levels). Given a shyness level $s \in [k]$, the audience is said to be s -insoluble iff there is a shyness level $s' \in [s]$ such that $\sum_{j \in [s'-1]} p_j < s'$. Otherwise, we say the audience is s -soluble.

The idea behind insolubility is the following. An audience which is s -insoluble contains less than s spectators who have shyness less than s . Therefore these guys will never stand up, thus blocking the guys with shyness level s . In particular, there won’t be a standing ovation for the prima donna.

The following Lemma gives a powerful necessary and sufficient condition for a standing ovation to eventually occur.

Lemma 1. There will eventually be a standing ovation iff the audience is k -soluble.

Proof: Indeed, “there is eventually a standing ovation” iff “for every shyness level $s \in [k]$ there are at least s spectators with shyness level less than s ” iff “ $\sum_{j \in [s-1]} p_j \geq s, \forall s \in [k]$ ” iff “the audience is k -soluble”. ■

B. The program proper

Consider the following short program:

1. **INITIALIZE** $s \leftarrow 1, r \leftarrow 0$.
2. **CHECK** If $\sum_{j \in [s-1]} p_j < s$, then
 3. **INVITE** a friend with any shyness level $s' \in [s]$.
 4. **UPDATE** $p_{s'} \leftarrow p_{s'} + 1, r \leftarrow r + 1$.
5. **UPDATE** $s \leftarrow s + 1$.
6. **CHECK** If $s = k$, then **RETURN** r . Else **GOTO** 2.

Theorem 2. The above program terminates after exactly k steps. Once it terminates, the resulting audience is k -soluble, and thus there will eventually be a standing ovation. Moreover, the program outputs the least number of friends to invite, namely $r(p_0, p_1, \dots, p_k)$.

We will need the following useful Lemmas for the proof.

Lemma 3. If the audience is s -soluble but $(s+1)$ -insoluble, then it becomes $(s+1)$ -soluble upon the invitation of a friend with any shyness level less than $s+1$.

Proof: Straightforward. Nothing to do. ■

Lemma 4. Define $0 < s_0 := \text{least } s \in [k] \text{ s.t. the audience is } s\text{-insoluble}$ ($s_0 := \infty$ if no such s exists). Then it holds that

$$\left. \begin{aligned} r(p_0, p_1, \dots, p_k) &= 0 \text{ if } s_0 = \infty, \text{ and } r(p_0, p_1, \dots, p_k) = \\ 1 + r(p_0, \dots, p_{s'-1}, p_{s'} + 1, p_{s'+1}, \dots, p_k) &\forall s' \in [s_0 - 1], \text{ else.} \end{aligned} \right\} \quad (1)$$

Proof: Indeed if $s_0 < \infty$, then by Lemma 3 inviting a friend with any shyness $s' \in [s_0 - 1]$ will simply subtract 1 from the least number of friends required to produce a standing ovation. This proves the first part of formula (1). On the other hand, if $s_0 = \infty$, then the audience is k -soluble, and thus (Lemma 1) will eventually produce a standing ovation on its own. ■

Proof of Theorem 2: Indeed, the program does nothing but compute $r(p_0, p_1, \dots, p_k)$ via the formula (1) established in Lemma 4, word-for-word. Also, by construction, it halts after exactly k steps and its output r is at most k . We are done. ■

a) *Example attaining the bound:* $r(0, 0, \dots, 0, p_k) = k$.

REFERENCES

- [1] Google: Standing Ovation, Problem A, Google code jam. (2015), <http://code.google.com/codejam/contest/6224486/dashboard>